

Question 1

Correct

Mark 100.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |



Buatlah implementasi kelas **BunchOfKeys** dalam bahasa pemrograman C++ yang memiliki sebuah atribut **n_keys** bertipe integer yang menyimpan jumlah kunci yang ada. Kelas juga **hanya** memiliki 2 buah *member function*, yaitu:

1. **add**: menambah kunci (hanya melakukan increment pada atribut **n_keys**)
2. **shake**: mengeluarkan bunyi "kringing" sejumlah **n_keys** kali (*setiap "kringing" diakhiri end-of-line*) apabila jumlah kunci > 1, atau mengeluarkan pesan "Tidak terjadi apa-apa" (*diakhiri end-of-line*) apabila jumlah kunci ≤ 1.

Pada saat konstruksi obyek, jumlah kunci semula adalah 0.

Diberikan header C++ sebagai berikut, Anda hanya diminta mengumpulkan file implementasi (**BunchOfKeys.cpp**) nya saja.

```
// BunchOfKeys.hpp
#ifndef __BUNCH_OF_KEYS_HPP__
#define __BUNCH_OF_KEYS_HPP__

#include <iostream>
using namespace std;

class BunchOfKeys {
public:
    // ctor
    BunchOfKeys();

    // member function
    void add();
    void shake();

private:
    int n_keys; // jumlah kunci yg ada
};

#endif
```

C++14

 [BunchOfKeys.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 25 | Accepted | 0.00 sec, 2.93 MB |
| 2 | 25 | Accepted | 0.00 sec, 2.87 MB |
| 3 | 25 | Accepted | 0.00 sec, 2.87 MB |

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 4 | 25 | Accepted | 0.00 sec, 2.89 MB |

Question **2**

Correct

Mark 100.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Diberikan file berikut yang berisi sebuah kelas bernama **Paper**. Kelas **Paper** menyimpan dua buah atribut integer dan menghasilkan keluaran khusus setiap dilakukan pemanggilan *constructor*, *copy constructor*, *copy assignment operator*, dan *destructor*.

1. [Paper.hpp](#)
2. [Paper.cpp](#)

Perhatikan contoh program berikut:

```
#include "Paper.hpp"
int main() {
    Paper a('A');
    a.fold();
    a.fold();
    return 0;
}
```

Program di atas jika dijalankan dengan kelas Paper yang disediakan, akan mengeluarkan output:

```
ctor A
fold A(1)
fold A(2)
dtor A
```

Sebagai seorang mahasiswa yang telah memahami konsep dasar OOP dengan baik, anda ditugaskan untuk membuat **program utama** dengan memanfaatkan kelas **Paper** yang mengeluarkan output sebagai berikut:

```
ctor A
ctor B
ctor C
cctor C
fold A(1)
fold B(1)
fold C(1)
glue C
fold C(2)
dtor A
setName C => X
fold C(1)
glue C
fold C(2)
dtor C
dtor X
dtor B
```

Unggahlah program utama tersebut dengan nama file **main.cpp**

Hint : Untuk melakukan dealokasi secara dinamis manfaatkan **pointer** dan fungsi **delete** untuk dealokasi pointer pada c++

Contoh sintaks delete:

```
#include "Paper.hpp"
int main() {
    Paper *a = new Paper('A');
    delete a;
    return 0;
}
```

```
}
```

C++14 ↕

 [main.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 100 | Accepted | 0.00 sec, 2.88 MB |

Question 3

Correct

Mark 100.00 out of 100.00

| | |
|--------------|-------|
| Time limit | 1 s |
| Memory limit | 64 MB |

Pak Ganesh akan berjualan panganan-panganan. Untuk saat ini, dia akan menjual keik dan panekuk sebagai kudapan malam. Uang yang diperoleh dari menjual keik adalah 51, sedangkan uang yang diperoleh dari menjual panekuk adalah 37. N_{rumah} adalah variabel untuk menghitung jumlah kunjungan Pak Ganesh ke rumahnya. Jumlah uang dan n_{rumah} mula-mula adalah 0.

Terdapat beberapa operator yang perlu diimplementasikan:


1. Operator $+$: Pak Ganesh tentunya akan menambahkan stok panganan yang dia jual. Dia akan pulang ke rumah dan menambahkan stok keik dan panekuknya. Setiap mengunjungi rumah, n_{rumah} akan bertambah sebanyak 1.
2. Operator $-$: Pak Ganesh akan menjual panganan yang dia miliki kepada pembeli. Dari penjualan tersebut, dia akan meraih keuntungan. Jika permintaan pembeli melebihi stok, Pak Ganesh hanya akan menjual sesuai dengan stok yang dia miliki.
3. Operator \wedge : Pak Ganesh akan menyumbangkan panganan yang dia miliki masing-masing sebanyak n buah. Jika ternyata stok yang dia miliki, baik keik maupun panekuk kurang dari n , maka uang yang dia peroleh akan berkurang seturut dengan kekurangan dari sumbangan n buah tersebut. Asumsikan uang dapat negatif (berhutang). Misal, Pak Ganesh ingin menyumbang $n = 6$, akan tetapi stok Pak Ganesh yang dimiliki adalah 3 keik dan 8 panekuk, maka uang Pak Ganesh akan berkurang $(6-3) * 51$ dan stok Pak Ganesh sekarang menjadi 0 keik dan 2 panekuk.

Bantulah Pak Ganesh untuk mengimplementasikan header Panganan berikut.

Berikut file [Panganan.hpp](#)

Kumpulkan dengan nama file **Panganan.cpp**

C++14

 [Panganan.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict | Description |
|----|-------|----------|-------------------|
| 1 | 10 | Accepted | 0.00 sec, 2.88 MB |
| 2 | 10 | Accepted | 0.00 sec, 2.85 MB |
| 3 | 10 | Accepted | 0.00 sec, 2.87 MB |
| 4 | 10 | Accepted | 0.00 sec, 2.87 MB |
| 5 | 10 | Accepted | 0.00 sec, 2.88 MB |
| 6 | 10 | Accepted | 0.00 sec, 2.87 MB |
| 7 | 10 | Accepted | 0.00 sec, 2.86 MB |
| 8 | 10 | Accepted | 0.00 sec, 2.89 MB |
| 9 | 10 | Accepted | 0.00 sec, 2.98 MB |
| 10 | 10 | Accepted | 0.00 sec, 2.89 MB |